(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

PCT

(43) International Publication Date
14 March 2002 (14.03.2002)

(10) International Publication Number
WO 02/21345 A1

(51) International Patent Classification[7]: G06F 17/50

(21) International Application Number: PCT/US01/26009

(22) International Filing Date: 31 August 2001 (31.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/653,504 1 September 2000 (01.09.2000) US

(71) Applicant: **THE ARIZONA BOARD OF REGENTS ON BEHALF OF THE UNIVERSITY OF ARIZONA** [US/US]; 1717 E. Speedway Boulevard, Room 3205, Tucson, AZ 85721 (US).

(72) Inventors: **MADHUSUDAN, Therani**; 7578 Eventana Vista Ct., Tucson, AZ 85750 (US). **ALDERFER, Ronald, R.**; 51500 Orange Road, South Bend, IN 46628 (US). **CREMER, Dana, S.**; 51863 Hedge Lane, Granger, IN 46530 (US). **FLASK, David, J.**; 15554 Spring Meadow Lane, Granger, IN 46530 (US). **LONG, Sarah, A.**; 51857 Meadow Crest Drive, South Bend, IN 46628 (US). **VAIDYANATHAN, Ganesh**; 15662 Coldspring Ct., Granger, IN 46530 (US).

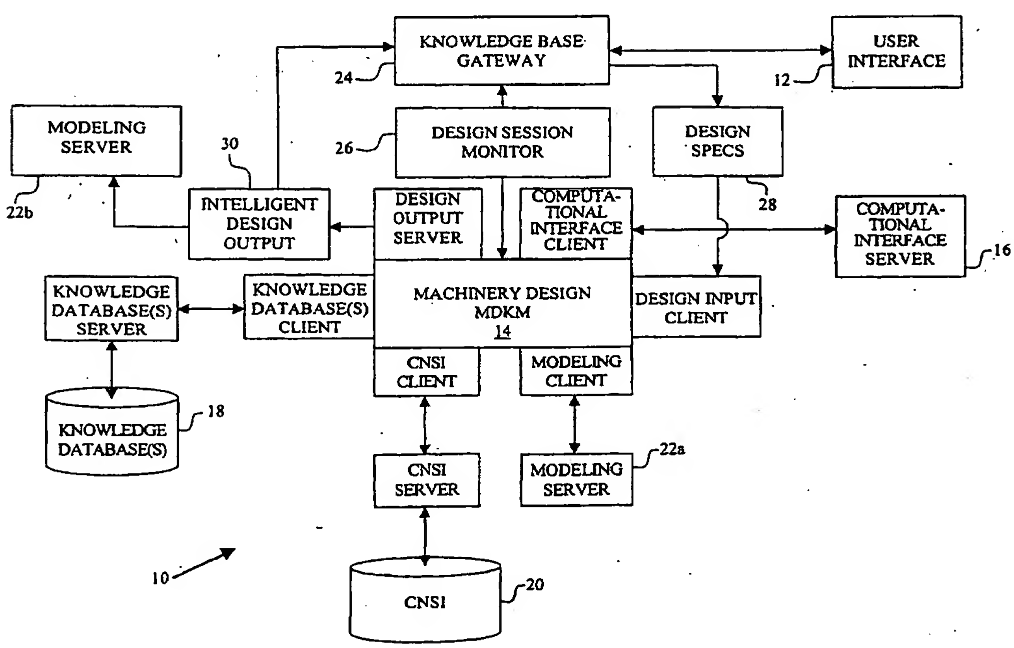
(74) Agent: **KUESTERS, Eckhard, H.**; Oblon, Spivak, McClelland,, Maier & Neustadt, P.C., 4th floor, 1755 Jefferson Davis Highway, Arlington, VA 22202 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,

[Continued on next page]

(54) Title: ADAPTATIVE MACHINERY DESIGN SYSTEM



(57) **Abstract:** A computer system (10) having an open process modeling architecture along with a structured knowledge acquisition framework used to intelligently and efficiently design machinery. The system has a machinery design knowledge manager (14) connected to a computational interface server (16) that together interact and coordinate the activities of one or more knowledge databases (18). The machinery design knowledge manager (14) executes the core decision making processes of system (10) and functions as a rule-based inference engine.

WO 02/21345 A1

CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

ADAPTIVE MACHINERY DESIGN SYSTEM

FIELD OF THE INVENTION

The present invention relates in general to software systems, including methods and apparatus, for designing complex mechanical equipment and, in particular, to a flexible and fully integrated adaptive software system suitable for designing sophisticated machinery and components thereof.

BACKGROUND OF THE INVENTION

Engineering design of complex mechanical systems is a loosely-structured, knowledge intensive and costly process. It is common practice in manufacturing industries for those involved in the design of a certain component of an equipment system to be effectively isolated from those charged with designing other components of the system. Aircraft wheel and brake design, for example, like any engineering intensive discipline, is an elaborate process involving many persons or teams using of a variety of conceptual and mathematical models, decision making processes and software tools. However, in this and many other industries, systemic inefficiencies exist which oftentimes greatly underutilize human and equipment resources. For instance, past product design and manufacturing data, whether or not deployed in successful product designs, might be stored on disparate or incompatible platforms or media, or worse, not retained in any tangible form. Under these conditions, potentially valuable sources of information may go untapped. Additionally, the many engineering personnel dedicated to designing the myriad components of complicated equipment systems frequently do not communicate in an integrated and concerted fashion, thereby resulting in duplicative or misguided efforts or other wastes of time and resources.

Engineering processes can be viewed as a sequence of decision making and validating subtasks that allow for the evolution of the product. A similar framework may be envisioned for numerous other decision making tasks in an enterprise. Within the context of automating

engineering processes to the fullest extent possible, technologies presently used to integrate design and manufacturing processes are commonly known as "Knowledge-based Engineering" or KBE. KBE is a collection of computing tools and techniques along with product and process modeling paradigms that may be applied to automate or streamline various decision making tasks in the design of a product. These collective computing tools typically incorporate two or more aspects of the classic design technologies listed below:

1. CAD (computer aided design) for solid modeling;

2. CAM (computer aided manufacturing) for rapid generation of manufacturing deliverables, i.e., numerical control data;

3. CAPP (computer aided process planning) for automating how a product will be manufactured;

4. Analysis - FEA (finite element analysis), classical analysis, etc.;

5. Manufacturing planning systems; and

6. Project management systems.

Most KBE tools facilitate the integration of a wide variety of these well-understood technologies to improve the overall design process. Judicious engineering the decision making involves the use of these tools along with as much product/process knowledge as possible to generate and evaluate numerous product alternatives and evolve the product from concept through detailed design. The vision of KBE is to integrate the data and decision-making in the different tools thereby guiding the process of product evolution from inception to product definition.

Numerous tools and approaches constitute KBE but no clear consensus definition exists in the industry. Further, these techniques are faced with some essential problems that have limited their wide- spread acceptance though the basic technologies have been around for approximately the past two decades. These problems include:

1. Integration of a wide variety of product data. For example, both geometric and non-geometric information needs to be processed and reasoned with during design. Development of a truly adaptable framework to allow modeling and capture of this data has been a major obstacle. Agreement on both semantics and syntax has been difficult across the mechanical CAD industry.

2. Integration of various reasoning algorithms. Decision making necessarily involves the use of a number of approaches to solve a single problem. The design process may involve numerous different kinds of steps from solving simple equations to looking up data in tables. This requires the need for new software modeling frameworks.

3. Engineering knowledge evolves constantly. A framework that allows for easy maintenance and incorporation of changes is very critical. The software should be highly evolvable and scalable.

4. Numerous other areas of an enterprise are being supported by complex software tools. KBE frameworks need to allow for sophisticated interactions between KBE tools and these external systems such as ERP (Enterprise Resource Planning) such that one may make better decisions with good exchange of data and processes between systems.

5. The technology must be adaptable to evolution in hardware and software technologies.

The first and second problems are effectively addressed by the development of a clean conceptual framework for product and process modeling. The latter three problems are best resolved by innovative use of modern object oriented technologies as a means to allow for incremental "tinker toy" building of complex software systems.

Among the currently available approaches to KBE is knowledge-based engineering for mechanical design. MCAD, as it is also known,

builds on a number of well-developed technologies from the fields of artificial intelligence (AI), software engineering, CAD, CAM, solid modeling, geometric reasoning, cognitive psychology, graphics, and others. The aim of MCAD is to provide a software system that mimics the process of design along with the entities being reasoned with. These entities constitute what is called the "product model" whereas the design activity is described by a "process model." A good KBE system requires development of both a flexible product and process model.

Issues that must be addressed in the development of a product model for KBE include comprehensiveness (how much of a product description does the model cover?), extensibility (can new elements be added to the model without requiring major overhauls?), semantics (what does each attribute and its values mean in the real world, can these attributes be computed/reasoned with?), syntax (how are the attributes organized, e.g., are colors encoded by numbers?), applicability (how much of the model is reusable and in what ways?), the nature of its implementation (object oriented versus flat files versus relational databases, etc.), and the different types of mathematical and non-mathematical models being integrated.

The process of design is a complex creative task. KBE systems vary in their ability to model processes of engine decision making and systems manage the complexity by allowing for iterations, branching, hierarchical nesting, and the like. The process modeling framework inside a KBE system determines the kinds of algorithms that can be called upon to operate on the aforementioned product model. The algorithms may include rule-based inference engines such as backward/forward chaining with the Rete Algorithm (an algorithm for the implementation of production systems), constraint satisfaction, searches, optimization, mathematical computations such as solving equations and so on. The algorithms mimic the processes of reasoning involved in the design

process. Extensibility of the process model is also an important capability as the design processes evolve or change. How quickly is the process model adaptable and how much is reusable are also significant concerns associated with KBE process model tools.

Various ways of building these systems and numerous research prototypes have been built addressing various aspects of KBE product model and process model frameworks. However, truly comprehensive commercial systems still do not yet exist. Historically, KBE systems have primarily originated from CAD or analysis companies with a focus on engineering, geometric reasoning or analysis and thus are limited in their capabilities by virtue of their basic bias toward one particular viewpoint or another. As such, they do not fully function as comprehensive integrators of both the product and process models. Accordingly, casting a design process for a product into a fixed kind of product or process model leads to less than desirable results. Design processes optimally involve a hybrid of reasoning techniques and perspectives. A need exists for KBE systems that readily incorporate changes in reasoning techniques, perspectives and other information as knowledge about the product and its design process evolves over the course of time.

A recently published summary of the state of the art of existing KBE systems is provided in *Knowledge-Based Engineering Systems: Applying Discipline and Technology for Competitive Advantage*, D. H. Brown Associates, Inc. (1999), from which the following observations may be made:

1. No KBE system currently exists that provides all generic design capabilities.

2. The features of these systems vary and are highly customizable.

3. The algorithms used in these systems vary considerably.

A commercially available KBE tool of which the present inventors are aware is The KBO Environment™ offered by Knowledge Technologies

International S.A. (KTI), a Luxembourg corporation, whose products are available on the World Wide Web at <URL:www.ktiworld.com>. That Web site indicates that the key pieces for The KBO Environment™, CAD technology and a so-called "blackboard" system were brought together by KTI's purchase of Black Board Technology Group (BBTech) of Amherst, Mass and BBTech's product, Generic Blackboard Builder. The following is a summary of the KTI approach to KBE tools based on information posted at their Web site and in the aforementioned *Knowledge-Based Engineering Systems: Applying Discipline and Technology for Competitive Advantage*, D. H. Brown Associates, Inc. (1999).

The algorithmic framework used by KTI is a process modeling engine commonly known as blackboard technology. Numerous blackboard approaches exist in the literature and are available, for example, in the handbooks of artificial intelligence. Within blackboard technologies as a class, numerous variants are possible for the modeling processes and their associated tools, managing execution control, and other factors.

Blackboard technology is a conceptual framework with two essential elements: knowledge sources and a blackboard. There is no overall predefined control logic. Blackboards support a framework called opportunistic problem solving wherein process steps execute in a very dynamic way, i.e., with no predefined order that depends on the state of the system. Further, the KTI system is intended to support what is called "distributed design" in a larger context. Modeling design processes as "distributed" is a complex endeavor. The KTI system, in particular, aims to coordinate various synchronous and asynchronous processes and does not appear to support hierarchical process modeling and recursive execution of the process steps in a predefined process hierarchy.

Blackboard based systems are useful for what is called exploratory programming and are well suited for computational problems wherein the knowledge is not very well defined (unlike in "mature" technologies,

discussed hereinafter) and the overall problem is being modeled in an incremental way. For well developed design processes such an approach may be costly to develop and deploy. These systems are highly expensive in terms of run time and hardware resources. Additionally, they are complex to initialize and maintain.

Some notions of KBE seem similar to ideas proposed by Parametric Technology Corporation <URL:www.ptc.com> of Waltham, MA called Behavior Modeling but the underlying needs that they address are very different. The purpose of Behavior Modeling is to do better solid modeling rather than the overall design per se. Although essential, solid modeling is a very small part of the overall KBE design task and the percentages of solid modeling as a core component of the design tasks varies considerably in the product being considered.

An advantage exists, therefore, for an equipment design system having an open and flexible software whose architecture is capable of facilitating integration of various software tools and decision-making processes, thus enabling unhindered information flow and overall automation of a desired design process.

A further advantage exists for a highly structured modeling and knowledge acquisition template that is capable of hierarchical process modeling and recursive execution of the process steps in a predefined process hierarchy for effectively capturing and assimilating the process of reasoning as well as engineering knowledge regarding a product and its components in order to generate new designs of similar equipment. Systemic integration and automation of the design process will enable mature manufacturing industries to have a framework to standardize their product development cycle, provide a means to capture the current knowledge that exists within the organization and reuse this knowledge to develop new products in shorter cycle times.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will become more readily apparent from the following description of preferred embodiments thereof shown, by way of example only, in the accompanying drawings wherein:

FIG. 1 is a schematic view of the increasing level of detail requirements that are encountered in the design of a typical aircraft landing equipment subsystem;

FIG. 2 is a schematic view of the function of a machinery design MDKM component of an adaptive equipment design system according to the present invention at a selected level of the machinery design process hierarchy;

FIG. 3 is a flow chart depicting process modeling that might be performed by the adaptive equipment design system according to the present invention when designing a selected machinery subsystem;

FIG. 4 is a schematic view of an adaptive equipment design system according to the present invention; and

FIG. 5 is a flow diagram depicting a representative equipment design process that may be conducted by the equipment design system according to the present invention.

SUMMARY OF THE INVENTION

The present invention offers a viable framework for building KBE systems with the latest developments in software and hardware technology that overcome most of the deficiencies of traditional KBE systems. The invention provides a computer system having an open process modeling architecture along with a structured knowledge acquisition framework that may be used to integrate software and processes that are currently employed in the design of sophisticated machinery. The process modeling architecture enables modeling of the

design process as a hierarchy of interacting subprocesses. Each subprocess models a specific decision making step involved in the design process. More specifically, each subprocess captures elements of a particular design step such as calculating a dimension, solving a mathematical system of equations, querying databases and choosing elements based on the query, logically trading off between multiple decision options, or running an external software application and interpreting the results.

The system possesses knowledge acquisition capability whereby historical and newly discovered design and manufacturing knowledge may be encoded into the system's software. The system may thus be used to capture years of knowledge concerning design and manufacturing methods and use that expertise as necessary to improve the design of desired products. Consequently, the system is especially well suited for designing mature engineered products having highly developed design histories and protocols, e.g., aircraft landing systems.

The software architecture of the KBE system of the present invention comprises a process engine, a product model and Web-based user interface. The process engine is a task execution engine that functions as an automatic scheduler that schedules activities in an order and then executes those activities as if in a general design project. To achieve desirable results, the process engine must possess the following capabilities and characteristics:

1. A way to load the engine with tasks (activities).

2. Rules for selecting these tasks based on certain criteria. The essential criteria in the context of engineering design is the current state of the design of the engineered product and also the tasks that have been completed thus far. These rules are called "design process knowledge."

3. After task selection, the process engine executes the selected tasks.

Prior to loading the process engine with tasks, the tasks must first be defined. Each task has a name, certain resources (i.e., inputs it takes) and also certain outputs that emerge after the task is completed. Any process in general can be described as a sequence of such tasks. To facilitate the selection and prioritization of tasks, the present system employs an embedded, rule-based, forward chaining, expert system. Expert systems are a well established technology. Unlike the existing use of expert systems to code product knowledge per se, however, the present invention uses tasks to model the different steps in the design process and then models the knowledge of how to execute the design process as rules of task selection.

Additionally, a significant aspect of the architecture of the instant system is that it allows for abstract modeling of complex nested process hierarchies via decomposition of tasks into recursively lower levels of complexity. Such abstraction in a rule-based system facilitates the selection of tasks and the creation and combination of tasks of numerous and disparate types. More particularly, possible tasks that can be accommodated and implemented by the system may include simple computations, interacting with users, accessing databases, interacting with solid models, and so on. The typical machinery design process usually incorporates a variety of these tasks and the present system permits them to be fully integrated. In addition, the rules of task selection, i.e., what to do next in a design process, constitutes core experiential design knowledge possessed by engineers. Incorporating these rules into the present system enables the system to clearly and accurately mimic or model such knowledge.

Thus, the level of abstraction obtainable by the present system provides a means to comprehensively cover the design process in general. Further, the system provides a means to capture and store such knowledge and reuse it. In a larger context, the process engine can be

used to model processes of different kinds, i.e, not only those of design but also those of analysis, manufacturing and other decision making processes. Under those circumstances, each task, instead of acting on a product feature, would instead act on some other kind of data.

The product model is a feature-based breakdown of the product. Features are defined as units and in general ways related to the product modeling needs. Feature units can thus be extended in various ways depending on the product model. For example, the exemplary product model described herein has generalized templates which cover conventional engineering concepts and specific information related to aircraft wheels and brakes. To model products of different kinds, the general templates could again be used and extended if needed. Specific information about the new product would be assimilated into the system in order to customize the system for design of the new product.

The Web-based user interface is a dynamic applet that can be adaptively and automatically self-reconfigured. It collects input from the user, provides output, and its behavior is dynamically controlled by the backend process engine. System tasks may or may not be designed to interact with the user interface.

A machinery design process conducted by the present system may be envisioned as a plurality of interrelated subprocesses which are based on a set of modifiable and reusable templates that permit intelligent modeling of different aspects of the design process. The system uses these templates in such a way that a subprocess or the network of subprocesses is executed in a logical and integrated, yet flexible, sequence. The system thus mimics the best aspects of traditional machinery design but is unencumbered by the aforementioned disadvantages and inefficiencies of conventional design practices.

In contrast with conventional blackboard KBE systems which employ distributed design methodologies for exploratory programming design of relatively undeveloped technologies, the present system is focused more toward localized design processes for components of so-called mature technologies. In many mature manufacturing industries, machinery component design is very well developed and defined. Models and processes that may be well-defined for mature engineered products may not need all the resources that a blackboard KBE system may require.

Unlike a blackboard based system, the overall behavior of the present system is dependent on a design process hierarchy defined for the system. When the system starts up, it is initialized with a product model and a predefined hierarchy of processes. The system then starts executing the process steps by selecting them in a dynamically varying order. The task selection rules the execution of various steps of the process hierarchy and thus mimic a complex design process.

Although described specifically herein in connection with the design of aircraft wheels and brakes, the open process modeling architecture and knowledge acquisition capability of the present invention may be adapted to model virtually any type of machinery. Accordingly, as used herein, the term "machinery" shall be understood to include any machine, apparatus, device, equipment, mechanism or component(s) therefor whose design procedure involves knowledge-based decision making subprocesses The flexible architecture permits seamless integration of various software tools and data sources in a process-oriented manner, whereby the system may be used to design essentially any machinery.

Other details, objects and advantages of the present invention will become apparent as the following description of the presently preferred embodiments and presently preferred methods of practicing the invention proceeds.

DETAILED DESCRIPTION OF THE INVENTION

The following are well-understood technologies that are integrated into the adaptive machinery design system of the present invention. The present inventors believe it is the synthesis of these technologies in developing implementation and modeling approaches that constitutes the uniqueness of the instant framework.

1. Feature-based modeling. Numerous notions of "features" abound in industry. The present invention provides a highly flexible and generalized notion whereby product and/or process features may be defined based on the needs of the designer.

2. Process modeling. Tools exist for process modeling and the like but the modeling and inference engine technologies vary quite considerably. Likewise, the types of process tasks that can be modeled are highly variable. The present invention uses a hierarchical process modeling framework with the capacity to embed complex logic to activate tasks in a comprehensive, overall process sequence.

3. Workflow. All computational tools implement some sort of workflow. Workflows that combine both manual interaction and automation vary in their degree of detail and capabilities. By making the present framework flexible enough to handle both manual user input and also run autonomously, the framework allows building of complex workflows interleaving both manual steps and automation tasks.

4. Modeling knowledge/knowledge management. This is a design and manufacturing industry term having numerous implementations and definitions. The present invention involves dealing with very complex knowledge and is unlike data warehousing and similar approaches. In data warehousing, a copy of transaction data is specifically structured for querying and reporting. Examples of data warehousing include relational databases, flat files and the like. It involves the use of deep engineering

models which are mathematical models of engineering interactions on phenomena.

Philosophically, the KBE system of the present invention has kept the open nature of both the product and process model in mind and has developed a very neutral, extensible framework. The software system can be viewed as a "toolkit" of a process/product model and the associated software for digitizing them.

Some basic notions that form the underpinnings of the present approach are:

1. A flexible "feature-based" model for product modeling. Product related information is captured in terms of knowledge units called "features". Features can be any conceptual entity that is involved in the process of reasoning to describe any information about the product. The notions of feature units are not constrained by any perspective such as geometry or analysis but can be developed based on the functional requirements of the entity. For example, one could have design feature units along with analysis feature units. Further, one may have features that are hybrids of these features. Also, it must be noted that a single feature type could support various types of algorithmic reasoning ranging from being used in a rule-based inference engine to being used as a data element in a row in a database. This wide spectrum of representation allows for the product model to be enhanced in various ways depending on the application of the KBE technology.

2. A modular "tinker toy" framework for process modeling. The present invention models a process at multiple levels of increasing detail in terms of subunits called "tasks." Tasks can be decomposed into recursive levels of more complex tasks with complex patterns of execution such as iterations, branches, etc. Further, these tasks are reusable entities and encapsulate different parameterized algorithmic

approaches or decision-tree approaches. In contrast to currently available engines, the process model is data-driven and can be easily and adaptively extended during run-time. New tasks can be added with minimum effort, execution sequences can be reorganized quickly and the overall KBE software can be evolved in a manageable manner without compromising product knowledge or the performance of the design process.

As described in greater detail later herein, each process step might employ a number of different computational techniques. The process modeling engine forms the core of the total integration approach of the present system. Combining the process model and the product model in one integrated manner that can perform useful computations enables the aforementioned synthesis of various design technologies and the synergistic results that flow therefrom. The following represents a specific application of the present system as it might be used in the design and manufacture of aircraft wheels and brakes, a so-called mature technology with a long prior history of engineering knowledge upon which to draw information.

FIG. 1 illustrates the product breakdown tree that might be expected for a conventional aircraft landing system. In an actual product manufactured by Honeywell Inc. of South Bend, IN, this tree has 10 subsystems, nearly 200 components, 1000 features and nearly 3000 dimensions.

The wheel and brake product is divided into subsystems that provide some functionality of the brake. For example, the carbon heat stack subsystem provides an energy absorption function, the piston adjuster system provides the function of engaging and disengaging the brakes, the torque tube provides the structural function of taking up rotary loads, and so on. Each subsystem is then divided into its constituent components that make up the subassembly, e.g., the rotors,

stators and other components of the carbon heat stack subsystem. A subsystem may consist of various alternative configurations of a given set of components (similar to how a computer is assembled, except that the combinations are extremely complex and frequently need to be evaluated in detail). Each component is then divided into a set of geometric and non-geometric features. The geometric features identify specific spatial, structural functions of that geometry on the part. For example, a thread feature provides fastening, a land feature may support a seal, slots or holes may accommodate other structural features, etc. Thus, components are made up of essential geometric features.

Non-geometric information such as material, tolerances, analysis parameters and coefficients and the like also make up features that describe the component. A component thus is defined by a feasible set of features from which a subset is selected in a design configuration. It may not be possible to have some features exist together. Geometric features are then finally made up of actual dimensions, e.g., length, width, thickness or diameter that are to be made during manufacturing of a part. These dimensions define in detail the actual sizes and shapes of all of the geometric entities. Along with the creation of ensuing drawings from different perspectives (which are reusable), this systematic breakdown helps in developing a clear, well-decomposed solid model (which are also reusable) based upon a well-defined product nomenclature. This product breakdown thus enables parametric resizing, reuse of old components and smaller pieces, and easy redesign of the overall product based on changes in requirements and other design constraints.

Given this breakdown of the product in terms of feature and dimensional entities, the next step is to capture the relations between features and their dimensions. Since features coexist on a component, there are geometric and analytical relations between these dimensions.

Formulating these relations and associating them with specific parameters and their combinations is perhaps the most significant task of knowledge acquisition. During the knowledge acquisition process, one also collects other information, such as tolerance, surface finish and the like that is involved in the detailed design of the product. Further, some component features may be analyzed with complex analytical models and the modeling knowledge is also captured by the system according to the invention. This knowledge is documented in what are commonly known as IPI forms which are a set of forms which function like task descriptions in a project. There are input, description, and output fields that capture all the entities that are required for the task. The process description may be a simple or complex computation or a simple database lookup, as illustrated in FIG. 3 and discussed below.

Once the basic product related information is acquired, the next step is to develop a scheme whereby the various computations required during design can be automated. Based on the hierarchical process model structure presented earlier, a process breakdown that complements the product breakdown is developed.

FIG. 2 is a simplistic representation of the overall algorithm that controls the functioning of a later-described machinery design knowledge manager (MDKM) 14 at one level of the process hierarchy.

At startup, the system is loaded up with a complete definition of a "tree-like" product model with nodes having empty values. At the end of the design process, all nodes have values. After the product model is loaded, the system begins implementing the top level process tasks and then subsequent tasks according to certain task selection rules. A task may be a leaf level task of the tree or it may point to a lower level of tasks within the tree. The MDKM walks through the tree based on the task selection rules. Those rules take into consideration information such as which tasks have been completed and what is the state of the product

model, i.e., which dimensions have values and which do not. When a task points to lower level of tasks, the system recurses down to that level and starts executing those tasks. When lower level tasks are completed, the system returns to the task at the next higher level. This process may be visualized as walking down a flight of stairs, then back up, then back down, and so on, depending on the nature of the process, repeating already traversed steps, backtracking etc.

More particularly, the MDKM initializes with a task pool of top level tasks. A startup task is predefined and is selected as the first task to execute. When the startup task executes, it may change the product model(some node may have a value based on some inputs) or it may execute without impacting the product design. When the startup task is completed it tells the MDKM it is completed and is removed from the task pool.

The task selection rules then select the next task to be executed based on current state of the product model. The newly selected task is then executed. When the task is executed, it may be a simple leaf task such as calculating a dimension or it may point to a lower layer of tasks as reflected in FIG. 3. If a new layer of tasks exists, the MDKM recursively creates a new embedded manager for that level of the tree and starts processing those tasks. When the task is completed, the MDKM deploys its task selection rules to select the next task to be performed and the cycle is repeated until there are no more tasks in the task pool, at which point the system stops.

This computational engine is built using an expert rule based system called Java Expert System Shell (JESS) and forms the core of the MDKM. JESS is a free expert system software tool written in JAVA and script compatible with the C Language Integrated Production System (CLIPS) expert system developed by the National Aeronautic and Space Administration. Generalizing this core ability of the MDKM is possible by

assimilating tasks of various types and implementing tasks in ways that make the system extremely adaptable. More particularly, the rule based system according to the invention accommodates:

1. Product related tasks
2. Non-product related tasks such as presenting data to a solid modeler, obtaining data from the user, etc.
3. Repetitions of the same task sequences in an iterative way.
4. Spawning and running tasks in parallel and on multiple machines.
5. Individual tasks that themselves can be complex algorithms that work on the same product data that has been modified by a previous step.
6. Tasks that request manual input from users on an as-needed basis and that also provide outputs in a dynamic manner. This allows manual interaction to be facilitated whenever needed.
7. Tasks that can create new tasks and add to the task pool thus making the system extremely dynamic and reactive in nature.

FIG. 3 illustrates a product tree which defines a hierarchical nesting of processes or tasks required for designing a hypothetical piece of machinery. The top level process is a breakdown of the overall design process into sequential stages of preliminary design followed by design retrieval, detailed design and then analysis. Once this process is completed, a finished design is obtained which may be visualized in Pro/ENGINEER or processed further. Pro/ENGINEER is multifunction product modeling and design software marketed by Parametric Technology Corporation of Waltham, MA. Whenever certain conditions are not satisfied, the system may loop back to the earlier stages and proceed again. Detailed design is built-up from the design processes or tasks for several subsystems.

FIG. 3 shows the design of just one subsystem, hence there is no subsystem level logic sequence depicted therein. Subsystem design processes are built up from component design tasks (denoted as C). Components are designed from a sequence of feature level tasks (denoted as F) which are made up of dimension level computational tasks (denoted as D). Each task may be a database lookup, a mathematical computation or use of any kind of reasoning algorithm to enable computation. The tasks are sequenced based on the relations obtained during the product modeling process mentioned above. The sequence embeds various kinds of conditional logic and rules that guide the design process. By traversing the tree of tasks from start to finish, including performing any analysis tasks (denoted as A), a completely finished product subsystem configuration will be generated at the end of the process.

In addition to its many other advantages, the software architecture of the present invention allows for modification of a given process in an incremental way by simply adding new tasks, deleting unnecessary tasks, reorganizing the task trees and so on. New features and components can also be manually added and deleted and the tree structure reorganized as needed. Moreover, all of the tasks also have access to the same product model that is evolving in time as tasks execute. In this way, the product model can be adaptively and automatically modified during the design process.

Preferably, the architecture is extremely scalable, whereby product/process trees of large sizes can be modeled. Further, it is desirable that the architecture be portable and capable of being run on a variety of software systems such as Windows®, UNIX, etc. Further, it is preferable to parallelize such that pieces of a tree can be run on multiple machines that work together on a common design problem. From an engineering design perspective, this framework suits engineering design

very well because it supports the hierarchical nature of product models in general.

Referring to FIG. 4 there is shown a computer system constructed in accordance with a presently preferred embodiment of the invention and identified generally by reference numeral 10. System 10 preferably functions as a shell and comprises a flexible and adaptable client-server architecture that employs any suitable object-oriented programming language such as, for example, JAVA or C++. System 10 may operate on any electronic communication network capable of enabling interactive participation by users of the system. Examples of communication networks that may support system 10 include, the Internet, a proprietary network, a local or wide area network, a wireless network, a telephone network, etc. By way of illustration but not limitation, system 10 may be a World Wide Web (Web) based system functioning on the Internet. Among other things, system 10 comprises one or more user interfaces 12 through which scientists, engineers and/or technicians skilled in the art of machinery design may communicate with a machinery design MDKM (MDKM) 14 and a computational interface server 16. MDKM 14 and computational interface server 16, in turn, interact with one another and coordinate the activities of one or more knowledge database(s) 18, a communication network systems integrator (CNSI) 20 for facilitating communication and data transfer on the communication network, and one or more proprietary or commercially available software programs or servers 22a and 22b for modeling machinery.

User interface(s) 12 may be any types of computer, e.g., mainframe, workstation or personal computer, capable of interactively participating in system 10. MDKM 14 executes the core decision making processes of system 10 and functions on a rule-based inference engine. A rule-based inference engine is an algorithmic method used to make or derive inferences from a given set of initial facts about a problem by

applying a set of "rules" that encode relevant knowledge that may brought to bear upon the problem. Such a system may be used for tasks such as scheduling activities (whose facts may include descriptions of activities and time durations, and whose rules may be conditions for sequencing and overlapping these activities) or configuration of computers (whose facts may be the computer components and their descriptions, and whose rules include the relations and constraints between these components as to their compatibility, location and the like). In system 10, the rule-based mechanism is used for scheduling and managing the steps of the design process and for providing reasoning regarding the components of the machinery to be designed as the design process is executed.

As described in greater detail below, MDKM 14 functions in the capacity of a client vis-à-vis the computational interface server 16, knowledge database(s) 18, CNSI 20 and at least one of the machinery modeling servers, in this case server 22a. Conversely, MDKM 14 functions as a server for modeling server 22b.

A presently preferred computational server 16 is Mathematica® computational software marketed by Wolfram research, Inc. of Champaign, IL and implemented in JAVA and C. The computational server supports numerous mathematical functions that may be encountered by MDKM 14 in the course of execution. MDKM 14 uses the server as a service passing requests and receiving back answers and proceeding with the necessary computations.

Knowledge database(s) 18 may include any historical information related to the design and/or manufacture of the systems, subsystems and components of the machinery chosen for design or redesign. Such information may include, without limitation, human experiential lessons learned about the machinery, digitally stored solid models of the machinery, customary and/or regulation-imposed design standards, notes,

process maps that describe tasks in the process of design and their interactions as a visual graph (wherein nodes are the tasks and arcs denote relations), dependency maps that describe attributes of a product (e.g., dimensions) and the relations between them, past designs of the machinery (whether or not successful) and IPI forms.

CNSI 20 may be any communication network services integrator appropriate for the communication network within which system 10 is implemented. For example, in a web based environment, CNSI 20 may be the suite of user interface, program logic, data and web server applications marketed by Oracle Corp. of Redwood Shores, CA.

As mentioned above, servers 22a and 22b may comprise proprietary or commercially available software programs for modeling machinery and components thereof. Server 22a desirably communicates directly with MDKM 14. Presently preferred solid model mechanical design software suitable for use as server 22a is Pro/ENGINEER. As an alternative to Pro/ENGINEER, server 22a may be an ICAD-type application. ICAD is a comprehensive feature-based geometric modeler which employs a geometric reasoning engine. The KBE system product model of the present invention can be extended to add this or a similar geometric engine by using commercial geometric kernels available from Spatial Technology Inc. of Boulder, CO or Unigraphics Solutions Inc. of Ann Arbor, MI.

Server 22b preferably comprises software capable of evaluating, under simulated "real-world" conditions, the performance of solid models generated by server 22a alone or in conjunction with other information available on knowledge database(s) 18. Presently preferred structural, thermal, motion and fatigue simulation and analysis software suitable for use as server 22b is NASTRAN/PATRAN. NASTRAN is marketed by MSC Corp. of Texas.

Each user interface 12 interactively communicates with a knowledge base gateway 24. Knowledge base gateway 24 preferably is a webserver based middle-ware that coordinates access to the MDKM 14. The webserver coordinates the interaction with a user in a push approach collecting input from the user as needed and delivering outputs as and when they are computed. The gateway delivers applets and html files and collects information from the user using a proprietary user interface technology. The knowledge base gateway 24 further communicates with a design session monitor 26 which monitors the activity of MDKM 14 and reports its observations to the knowledge base gateway. The design session monitor is a software entity embedded in the MDKM 14 that keeps track of the state of execution and history of the MDKM 14 allowing for user interaction and guidance for the design process.

When functioning in its role as client, MDKM 14 queries, either directly or through various servers (as illustrated), computational interface server 16, knowledge database(s) 18, CNSI 20, modeling server 22a and knowledge base gateway 24. MDKM 14 may function in manual or guided design modes. As discussed more fully below, when operating in "manual" mode, the user interactively navigates through the design process. Each step of the process is presented visually to the user and the user may execute the computation and then proceed to the ensuing step. In manual mode, the system functions as a tutoring engine leading the user through the steps of a complex design process in an orderly manner. However, MDKM 14 does not execute any computations unless requested by the designer. This mode is especially useful for testing and validating the system as well as training new designers in the process of design.

In manual mode, the user instructs MDKM 14 to select a particular process for the design of a particular component or subsystem. System 10 then provides the starting step followed by the next step, thus

ensuring a complete walk through of the design process. During the manual process, the system and the user interact only at specific steps and the intelligence is provided by the user. The system functions in a passive mode receiving instructions from the user.

In the "guided" mode, the MDKM actively functions to guide the user through the design process using information from knowledge database(s) 18 that appear to satisfy the user-defined design criteria. When functioning in its role as server, MDKM 14 supplies intelligent design output 30 both to the modeling server 22b, whereby it may perform its simulated product performance modeling functions, and to the knowledge base gateway 24, whereby it may be observed by the user via user interface 12.

Referring to FIG. 5, there is shown illustrative, but not limitative, examples of how system 10 may be used to design sophisticated machinery, namely, subsystems of aircraft landing systems.

The system architecture described herein, including the front end (inputs) and back end (server architecture and outputs), may generally apply to design processes for any machinery. For example, the general structure of the system 10 can be applied to design of turbo machinery, engines and countless other types of equipment although the specific design data and steps may vary at the level of subprocesses. The specific information that may be provided in these steps would dictated by the machinery to be designed. In other words, depending on the machinery to be designed, more or less information may be provided or steps executed in the design process than exemplified herein in connection with aircraft landing gear.

To begin the overall design process, the user, through user interface 12, invokes the knowledge base gateway 24 and obtains the front page (or home page) 32 of the design system 10. At this location,

the user may opt to view the overall system to observe its capabilities and operational status. Alternatively, the user may choose to browse the system at 34 to review, for example, customary or regulation-imposed design standards associated with design of the subject machinery or its components, empirical design knowledge or past designs. At 36, the user may search and query one or more of the knowledge database(s) 18 to obtain any of an assortment of information that may useful or appropriate for a particular piece of machinery or component under consideration. Typical searchable design information that may be relevant to the design/manufacturing process may include human experiential lessons learned, electronically stored solid models, customary or regulatory design standards, notes, IPI forms, maps, past designs, etc.

As indicated at 38, once at front page 32, the user alternatively may choose to commence design of the desired machinery and/or component(s) thereof. More particularly, the user may elect either of "manual" or "guided" modalities by which to design the equipment, as respectively indicated by reference numerals 40 and 42. In "manual" mode, the user completely controls which information is retrieved from the knowledge database(s) 18 and how that information is used and processed. For example, as shown at 44, the user initially selects which equipment, system or component(s) thereof that are to be designed. In the case of aircraft landing equipment, the user may choose to design elements of the wheel assembly, nose wheel assembly or brake assembly systems. Within each selected system, the user may select a desired subsystem, e.g., wheel bearings, outboard wheel, inboard wheel, wheel bolt joints, torque tube, steel heat stack, piston adjuster, piston housing, carbon heat stack and brake bolt joints. Lastly, the user may choose a specific component or components form each subsystem as candidates for design or redesign. By way of example, typical aircraft landing equipment may be comprised of more than one hundred individual components that may be designable by system 10. Other sophisticated

machinery may involve less or considerably more components that may be subject to design by system 10.

Continuing at 44, following selection of the objects to be designed, the user may, without limitation, view system maps, navigate IPI forms or view a desired subsystem map. Having gathered information believed to be useful, the user may then manually execute the design process by using the MDKM or performing the computations manually. In manual mode, the user interactively communicates with MDKM 14 in such a way that the user has total control over the decision making involved in the design process and uses the MDKM only when needed. MDKM 14 performs no independent design operations. That is, MDKM 14 participates as an interface medium for enabling communication between the user and the other design/modeling modules of system 10.

A new design is then created and reports, defined solid models and other useful output concerning the new design is generated at 46. The generated output, regardless if ultimately incorporated into a successful product design, may be routed by MDKM 14 to the knowledge database(s) 18 and stored therein for future reference. The newly acquired knowledge is preferably defined in terms of a set of standard templates and/or process graphs which are then encoded into the above software framework and stored in knowledge database(s) 18. If the design appears to be favorable, the generated output may be used in tool design and manufacturing process planning necessary for developing the new design into a prototype.

Returning to step 42, the user may alternatively choose the "guided" mode of operation by which MDKM 14 proactively contributes to the design of the selected equipment thus "guiding" the designer. If guided design is desired, the user is initially queried whether a new design or a modification of an existing design is preferred. Following this, the system at step 48 asks the user to select a desired functional area or

context for the design process. For instance, system 10 may ask whether the user wishes for the system to execute a preliminary design, a detailed design or an analysis of a design. At step 50, the user is queried by system 10 to then select which equipment system, subsystem and component(s) are to be designed. At this point, the user inputs design specifications including product features, customer requirements, subsystem requirements, interactive subsystem dependencies, rules of thumb, user options and other design constraints appropriate for design of the selected equipment. These design specifications constitute the initialized state of the aforementioned product model tree.

Responsive to the design specifications input by the user at step 50, the logic incorporated in MDKM 14 instructs the MDKM to begin the guided modeling process at step 52. At startup, the system begins implementing the predefined top level process (startup) tasks and then subsequent tasks according to certain task selection rules by setting system options, search choices, analysis choices and defaults hierarchically prescribed in the logic as described above in connection with FIGS. 2 and 3. Following execution of the startup tasks, the task selection rules then select the next task to be executed based on the current state of the product model. The newly selected task is then executed. When the task is executed, it may be a simple leaf task such as calculating a dimension or it may point to a lower layer of tasks as depicted in FIG. 3. If a new layer of tasks exists, MDKM 14 recursively creates a new embedded manager for that level of the tree and starts processing those tasks. When the task is completed, the MDKM deploys its task selection rules to select the next task to be performed and the cycle is repeated until there are no more tasks in the task pool, at which point the system stops.

Thereafter, as shown at 54, MDKM 14 accesses at least one knowledge database 18 and retrieves plausible designs. Through

knowledge base gateway 24 and design session monitor 26, the user may passively or actively interact with MDKM 14 in the design process, as shown at step 56. For example, the user may simply view the status of the activities of MDKM 14 or view the solid design model(s) selected by the MDKM. Alternatively, the user may actively modify the design options, and instruct MDKM 14 to select other plausible designs from knowledge database(s) 18. Still further, the user might reject all of the designs selected by MDKM 14 and manually interact with the several system architecture modules of system 10 to generate other possible designs. Once the user is satisfied with a new design, the user directs system 10 to produce the reports, defined solid models and other useful output about the new at step 46. Once again, the generated output, regardless if ultimately incorporated into a successful product design may be routed by MDKM 14 to the knowledge database(s) 18 and stored therein for future reference. And, if the design, whether created by the user or suggested by MDKM 14, appears to be favorable, the generated output may be forwarded to manufacturing, tool design and process planning for development into a prototype.

The software architecture of system 10 thus brings together a number of computational techniques which have been commercially used in the development of applications such as work-flow technologies, rule-based systems, object oriented modeling, etc. Among its unique features, however, is the synthesis of these techniques thereby producing a flexible process and data modeling environment. In addition, its knowledge acquisition capability uses common conceptual foundations such as feature-based modeling, process graphs and the like to meaningfully capture historical and newly generated engineering knowledge in the framework of the software architecture so as to enable utilization of the knowledge as an additional tool in the machinery design and manufacturing process.

The present system thus uniquely promotes rapid preliminary product design and facilitates product decision making and manufacturing definitions by affording a user with an integrated array of modeling tools and human experience data combined in one package.

Although the invention has been described in detail for the purpose of illustration, it is to be understood that such detail is solely for that purpose and that variations can be made therein by those skilled in the art without departing from the spirit and scope of the invention except as it may be limited by the claims.

What is claimed is:

1. A computer system for designing machinery comprising:

at least one user interface for enabling a user to interact with the system;

at least one modeling server for designing machinery;

a computational interface server for enabling operation of said at least one modeling server;

at least one knowledge database containing historical information related to at least one of design and manufacture of the machinery; and

a machinery design knowledge manager connected to said at least one user interface, said at least one modeling server, said computational interface server and said at least one knowledge database, said machinery design knowledge manager functioning on a rule-based inference engine and being operable to create designs of the machinery using said at least one modeling server and said computational interface server, said rule-based inference engine being operable to generate a product model of the machinery to be designed through execution of tasks, said rule-based inference engine including task selection rules comprising a predefined hierarchy of process steps whereby selection of a subsequent task from said hierarchy of process steps is dependent upon the current state of the product model following execution of a preceding task.

2. The system of claim 1 wherein said machinery design knowledge manager is operable to select at least one plausible design of the machinery from said at least one knowledge database responsive to generation of said product model.

3. The system of claim 1 wherein said tasks are selected from the group consisting essentially of startup tasks, dimension level computational tasks, feature level tasks, component design tasks and analysis tasks.

4. The system of claim 1 wherein the system operates on an electronic communication network and further comprises a communication network systems integrator connected to said machinery design knowledge manager, said communication network systems integrator being operable to facilitate communication and data transfer on said communication network.

5. The system of claim 4 wherein said communication network is selected from the group consisting of the Internet, a proprietary network, a local or wide area network, a wireless network and a telephone network.

6. The system of claim 1 wherein said at least one knowledge database includes at least one of human experiential lessons learned about the machinery, digitally stored solid models of the machinery, design standards for the machinery, notes, maps, past designs of the machinery and input process information forms.

7. The system of claim 1 wherein said at least one knowledge database is capable of storing designs of the machinery created on the system.

8. The system of claim 1 wherein the machinery is aircraft landing equipment.

9. The system of claim 1 wherein the machinery is a modification of an existing design.

10. The system of claim 1 wherein said system enables design of the machinery in a manual or a guided design mode.

11. A method of designing machinery comprising the steps of:

(a) selecting a computer system for designing machinery, said computer system comprising:

at least one user interface for enabling a user to interact with the system;

at least one modeling server for designing machinery;

a computational server for enabling operation of said at least one modeling server;

at least one knowledge database containing historical information related to at least one of design and manufacture of the machinery; and

a machinery design knowledge manager connected to said at least one user interface, said at least one modeling server, said computational interface server and said at least one knowledge database, said machinery design knowledge manager functioning on a rule-based inference engine and being operable to create designs of the machinery using said at least one modeling server and said computational interface server, said rule-based inference engine being operable to generate a product model of the machinery to be designed through execution of tasks, said rule-based inference engine including task selection rules comprising a predefined hierarchy of process steps whereby selection of a subsequent task from said hierarchy of process steps is dependent upon the current state of the product model following execution of a preceding task; and

(b) using said computer system to create designs of the machinery.

12. The method of claim 11 wherein said machinery design knowledge manager is operable to select at least one plausible design of the machinery from said at least one knowledge database responsive to generation of said product model.

13. The method of claim 11 wherein said tasks are selected from the group consisting essentially of startup tasks, dimension level computational tasks, feature level tasks, component design tasks and analysis tasks.

14. The method of claim 11 further comprising operating the system on an electronic communication network, said computer system

further comprising a communication network systems integrator connected to said machinery design knowledge manager, said communication network systems integrator being operable to facilitate communication and data transfer on said communication network.

15. The method of claim 14 wherein said communication network is selected from the group consisting of the Internet, a proprietary network, a local or wide area network, a wireless network and a telephone network.

16. The method of claim 11 wherein said at least one knowledge database includes at least one of human experiential lessons learned about the machinery, digitally stored solid models of the machinery, design standards for the machinery, notes, maps, past designs of the machinery and input process information forms.

17. The method of claim 11 further comprising storing designs of the machinery created on the system on said at least one knowledge database.

18. The method of claim 11 wherein the machinery is aircraft landing equipment.

19. The method of claim 11 wherein the machinery is a modification of an existing design.

20. The method of claim 11 further comprising searching said at least one knowledge database for information useful for designing the machinery.

21. The method of claim 11 further comprising generating output concerning a new machinery design that is useful to tool design and manufacturing process planning necessary for manufacturing machinery in accordance with the new machinery design.

22. The method of claim 11 wherein step (b) comprises using said system in a manual or a guided design mode.

23. A method of designing machinery comprising the steps of:

(a) selecting a computer system for designing machinery, said computer system comprising:

at least one user interface for enabling a user to interact with the system;

at least one modeling server for designing machinery;

a computational server for enabling operation of said at least one modeling server;

at least one knowledge database containing historical information related to at least one of design and manufacture of the machinery, said at least one knowledge database including at least one of human experiential lessons learned about the machinery, digitally stored solid models of the machinery, design standards for the machinery, notes, maps, past designs of the machinery and input process information forms; and

a machinery design knowledge manager connected to said at least one user interface, said at least one modeling server, said computational interface server and said at least one knowledge database, said machinery design knowledge manager functioning on a rule-based inference engine and being operable to create designs of the machinery using said at least one modeling server and said computational interface server, said rule-based inference engine being operable to generate a product model of the machinery to be designed through execution of tasks, said rule-based inference engine including task selection rules comprising a predefined hierarchy of process steps whereby selection of a subsequent task from said hierarchy of process steps is dependent upon the current state of the product model following execution of a preceding task; and

(b) using said computer system to create designs of the machinery.

24. The method of claim 23 wherein said machinery design knowledge manager is operable to select at least one plausible design of the machinery from said at least one knowledge database responsive to generation of said product model.

25. The method of claim 23 wherein said tasks are selected from the group consisting essentially of startup tasks, dimension level computational tasks, feature level tasks, component design tasks and analysis tasks.

26. The method of claim 23 further comprising operating the system on an electronic communication network, said computer system further comprising a communication network systems integrator connected to said machinery design knowledge manager, said communication network systems integrator being operable to facilitate communication and data transfer on said communication network.

27. The method of claim 26 wherein said communication network is selected from the group consisting of the Internet, a proprietary network, a local or wide area network, a wireless network and a telephone network.

28. The method of claim 23 further comprising storing designs of the machinery created on the system on said at least one knowledge database.

29. The method of claim 23 wherein the machinery is aircraft landing equipment.

30. The method of claim 23 wherein the machinery is a modification of an existing design.

31. The method of claim 23 further comprising searching said at least one knowledge database for information useful for designing the machinery.

32. The method of claim 23 further comprising generating output concerning a new machinery design that is useful to tool design and manufacturing process planning necessary for manufacturing machinery in accordance with the new machinery design.

33. The method of claim 23 wherein said system enables design of the machinery in a manual or a guided design mode.

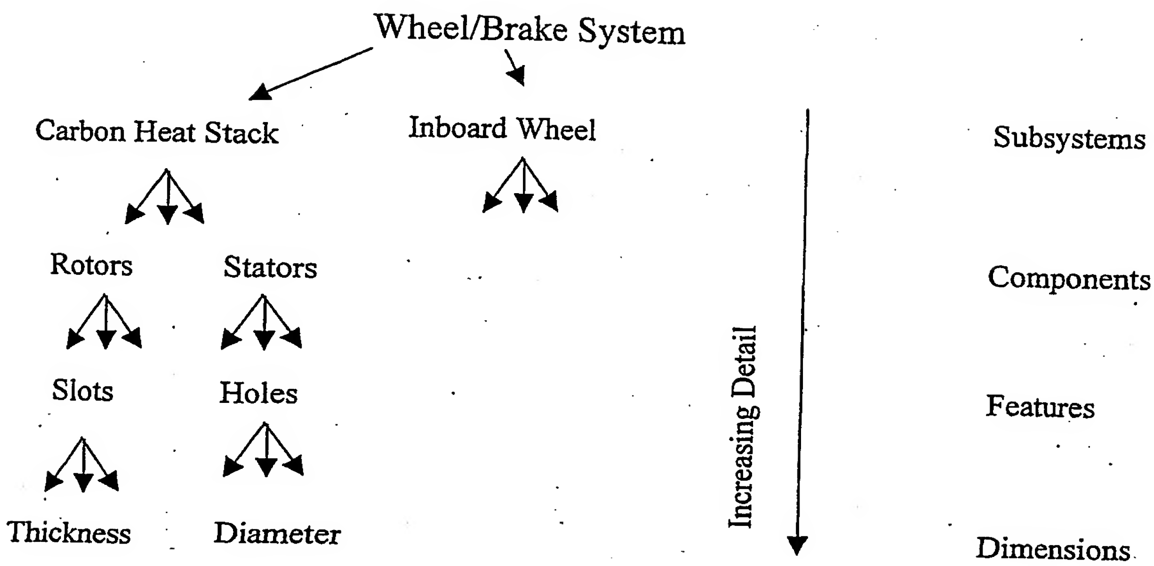

Wheel/Brake System
Carbon Heat Stack
Inboard Wheel
Rotors
Stators
Slots
Holes
Thickness
Diameter
Increasing Detail
Subsystems
Components
Features
Dimensions

FIG. 1

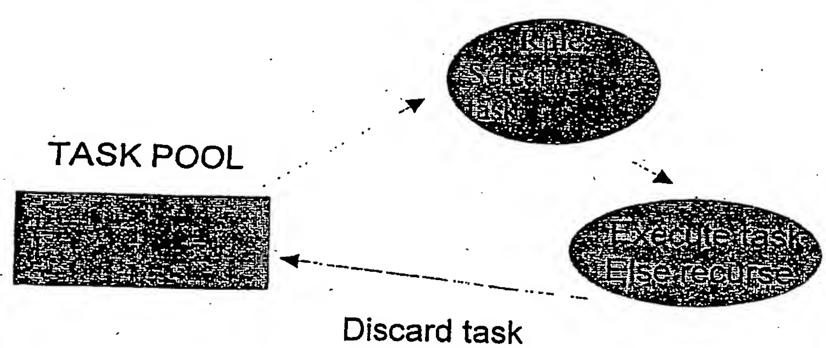

TASK POOL
Rules
Execute task
Else recurse
Discard task

FIG. 2

Subsystem Process Model

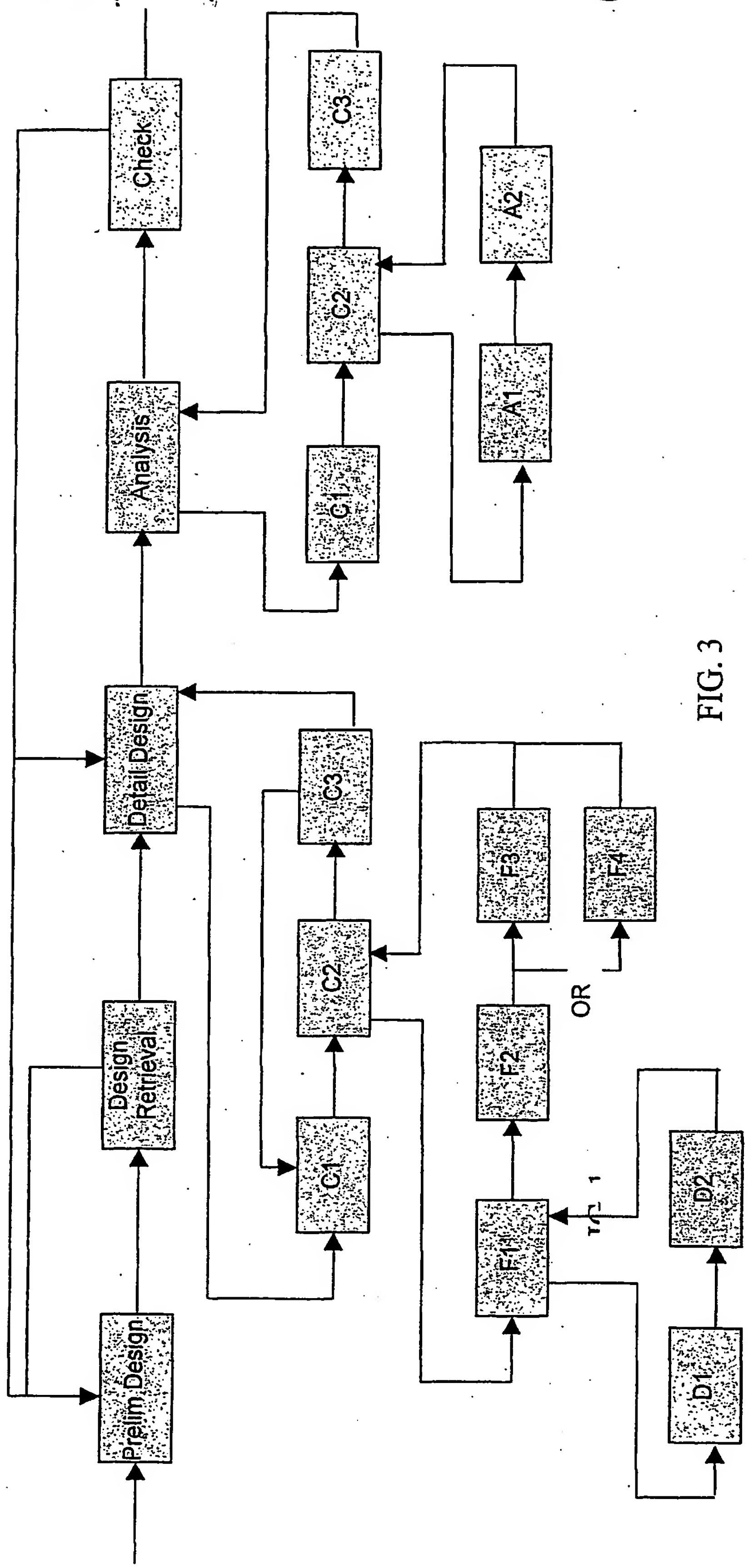


FIG. 3

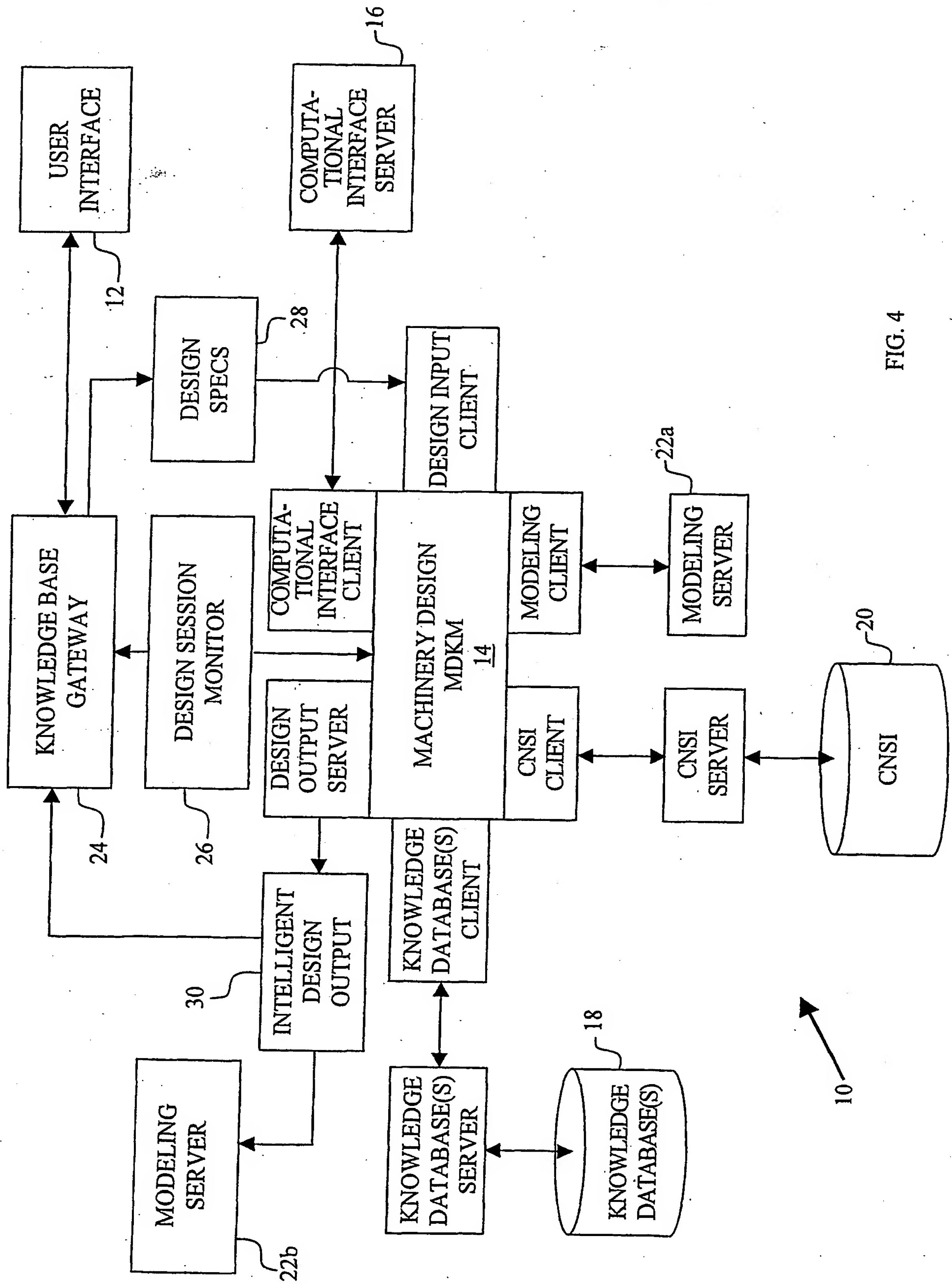

USER INTERFACE
12
COMPUTA-TIONAL INTERFACE SERVER
16
DESIGN SPECS
28
DESIGN INPUT CLIENT
KNOWLEDGE BASE GATEWAY
24
DESIGN SESSION MONITOR
26
COMPUTA-TIONAL INTERFACE CLIENT
MACHINERY DESIGN MDKM
14
MODELING CLIENT
MODELING SERVER
22a
DESIGN OUTPUT SERVER
CNSI CLIENT
CNSI SERVER
CNSI
20
KNOWLEDGE DATABASE(S) CLIENT
INTELLIGENT DESIGN OUTPUT
30
MODELING SERVER
22b
KNOWLEDGE DATABASE(S) SERVER
KNOWLEDGE DATABASE(S)
18
10

FIG. 4

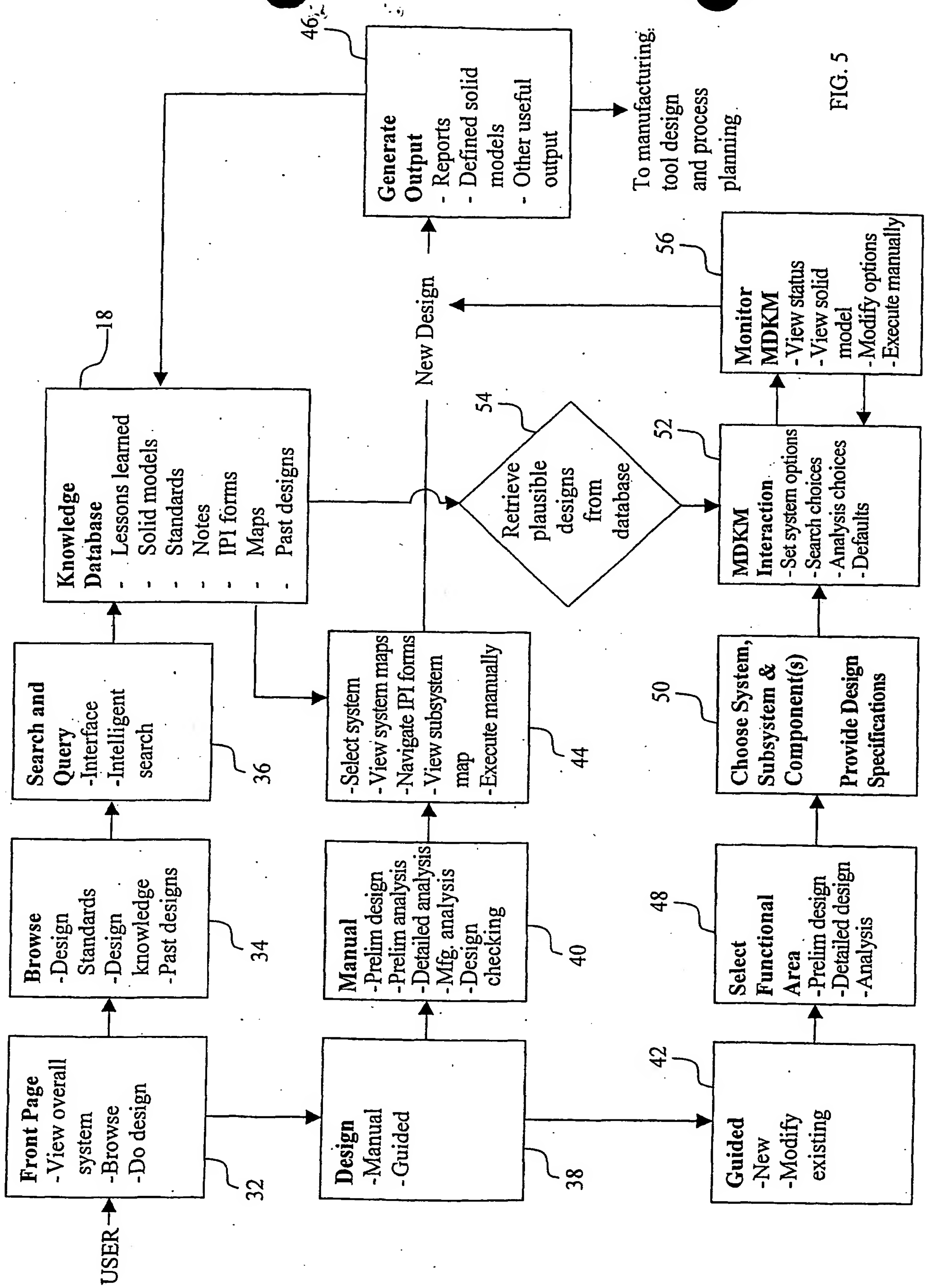

USER
Front Page
-View overall system
-Browse
-Do design
32
Browse
-Design Standards
-Design knowledge
-Past designs
34
Search and Query
-Interface
-Intelligent search
36
Knowledge Database
- Lessons learned
- Solid models
- Standards
- Notes
- IPI forms
- Maps
- Past designs
18
Design
-Manual
-Guided
38
Manual
-Prelim design
-Prelim analysis
-Detailed analysis
-Mfg. analysis
-Design checking
40
-Select system
-View system maps
-Navigate IPI forms
-View subsystem map
-Execute manually
44
New Design
Generate Output
- Reports
- Defined solid models
- Other useful output
46
To manufacturing tool design and process planning
Guided
-New
-Modify existing
42
Select Functional Area
-Prelim design
-Detailed design
-Analysis
48
Choose System, Subsystem & Component(s)
Provide Design Specifications
50
Retrieve plausible designs from database
54
MDKM Interaction
-Set system options
-Search choices
-Analysis choices
-Defaults
52
Monitor MDKM
-View status
-View solid model
-Modify options
-Execute manually
56

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/26009

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/50
US CL :703/1
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 703/1; 700/96-97, 116-118, 175-182

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, IEEE, ACM, Proquest
search terms: machine*, design, knowledge, task*, rule*

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	LANGER.G. et al. An Architecture for Agile Shop Floor Control Systems. Journal of Manufacturing Systems. 2000. Vol. 19 No. 4. pages 267-281.	1-33
A	SORMAZ.D. et al. Modeling of Manufacturing Feature Interactions for Automated Process Planning. Journal of Manufacturing Systems. 2000. Vol. 19 No. 1. pages 28-45.	1-33
A	ANONYMOUS. Controls, CAD/CAM, and Factory Automation. Manufacturing Engineering. August 2000. Vol. 125 No. 2. pages 194 et seq.	1-33

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 OCTOBER 2001	07 DEC 2001

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	SAMUEL BRODA James R. Matthews
Facsimile No. (703) 305-3230	Telephone No. (703) 305-1026

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/26009

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JONEJA.A. et al. Setup and Fixture Planning in Automated Process Planning Systems. IIE Transactions. July 1999. Vol. 31 No. 7. pages 653-665.	1-33
A	US 5,485,390 A (LECLAIR et al) 16 January 1996.	1-33